Modout: Learning to Fuse Modalities via Stochastic Regularization

Fan Li, University of Guelph, ON, Canada
Natalia Neverova, Facebook, Paris, France
Christian Wolf, LIRIS, INSA-Lyon, Lyon, France
Graham Taylor, University of Guelph, ON, Canada

Abstract

Model selection methods based on stochastic regularization such as Dropout have been widely used in deep learning due to their simplicity and effectiveness. The standard Dropout method treats all units, visible or hidden, in the same way, thus ignoring any *a priori* information related to grouping or structure. Such structure is present in multi-modal learning applications, where subsets of units may correspond to individual modalities. In this abstract we describe Modout, a model selection method based on stochastic regularization, which is particularly useful in the multi-modal setting. Different from previous methods, it is capable of learning whether or when to fuse two modalities in a layer. Evaluation of Modout on the Montalbano gesture recognition dataset demonstrates improved performance compared to other stochastic regularization methods, and is on par with a state-of-the-art carefully designed fusion architecture.

1 Introduction

Deep learning methods have proven effective on various multi-modal learning problems due to their ability to learn complex and useful representations in a domain-agnostic way [1, 2, 3]. However, fusing multiple modalities effectively is an unsolved problem. It is already well-known that good results are not likely to be achieved by simply concatenating features belonging to different modalities into a single "fully-connected" layer. Previous work has mainly focused on multi-modal analysis of RGB-D action videos [4, 5, 6]. For example, [6] proposed carefully designed multimodal layers for RGB-D object recognition, which fuse color and depth information by enforcing the transformed features to share a common part. [4] also attempted to discover the shared and informative components of RGB-D signals using a deep autoencoder-based nonlinear common component analysis. But the generalized performance of these methods to modalities beyond RGB-D videos is unknown. [3] explored the fusion of multiple modalities including RGB-D videos, mocap, and audio. They used a carefully-designed network architecture to gradually fuse modalities, and they found empirically that it is better to fuse modalities that have higher correlation (e.g., visual modalities first, then motion capture, then audio).

In the context of multi-modal gesture recognition, [3] introduced a Dropout-like regularization scheme called Moddrop. During training, Moddrop randomly removes the input from one or more modalities. This was shown, at test time, to improve robustness to corruption or loss of modalities. Our proposed Modout algorithm takes a different approach than ModDrop [3]. Instead of dropping the units belonging to a modality, in Modout the connections between the units in two adjacent layers are dropped with prior knowledge of modality-specific groupings. It has mainly two advantages over ModDrop. First, Modout can learn whether and when to fuse two modalities by optimizing the probabilities of dropping the connections between the two modalities. Second, Modout can be applied to any layer – not just the input layer. Although outside the scope of this abstract, Modout could, in theory, apply to other types of known groupings beyond modalities.

2 Related work

Regularization is a crucial component of training large neural networks, and advances in regularization have played a role in deep learning's advancement across large-scale applications. Traditional methods of regularization such as early stopping, weight decay, weight constraints, or addition of noise during training can be viewed as a means of limiting the capacity within a model and therefore its ability to overfit.

A new class of regularization methods that are stochastic have been widely used in deep learning due to their simplicity and effectiveness. At training time, these methods randomly remove certain structural elements of the network for each presented example, or collection of examples. The elements can be hidden or visible units (Dropout [7]), connections (DropConnect [8]), or even layers (stochastic depth [9]). At test time, the original network is used for prediction with a rescaling factor to compensate for the absence of elements during training. By pruning the network in a stochastic manner, stochastic regularization methods can be considered as a kind of ensemble that improves generalization via model averaging.

In the standard Dropout method, all units in a layer are dropped at the same rate, and therefore it ignores any structuring of the inputs which may result in more correlation among certain inputs. For example, pixels in an image are more correlated if they are spatially adjacent to each other. Also, for multimodal learning, there are more correlations for features within a modality. Recently, several variants of Dropout have been proposed which aim to exploit this correlation. Tompson et al. [10] proposed SpatialDropout for convolutional layers, in which adjacent pixels in the drop-out feature maps are either all dropped-out or all preserved. Neverova et al. [3] proposed ModDrop for multimodal learning, in which the input features belonging to the same modalities are either all dropped-out or all preserved. These methods have been shown to outperform standard Dropout, while their drop-out rates are pre-defined hyper-parameters.

[11] have recently proposed a method of learning the structure of deep neural networks via deterministic regularization. They insert, between each pair of fully connected layers, a sparse diagonal matrix whose entries are l_1 penalized. This implicitly defines the size of the effective weight matrices at each layer. The approach has a similar effect to Dropout.

An exception to the Dropout-variants is Blockout [12], which is also very related to our work. Blockout generalizes Dropout by introducing cluster assignments for each unit. Both the (implicit) dropout rates and the parameters are learned using backpropagation. Similar to Dropout and DropConnect, Blockout does not use the information regarding structural groupings among units, and the number of clusters needs to be set and tuned. Instead, at every layer, Modout ties the clusters to the modalities, and only learns the probability of fusion between each pair of modalities. The result is a substantial reduction in number of free parameters and one less hyperparameter to tune.

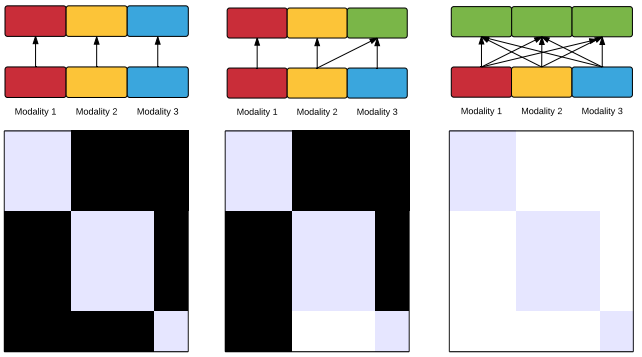
3 Model description

Most stochastic regularization methods can be considered as applying a stochastic mask to the weight matrix. The proposed Modout method shown in Fig. 1 is similar to Blockout in the sense that units are assigned to clusters. But instead of generating the cluster assignments randomly, the clusters in Modout are assigned based on knowledge of modalities. Therefore, in our case, each unit is assigned a unique cluster label, while in Blockout units can be assigned to more than one cluster. The number of units that belong to each modality in a hidden layer can be set to be proportional to the number of features in the input layer if it is not otherwise specified. During the entire process, the cluster assignments for all the units remain the same. Different from Blockout, which learns the probability of assigning units to clusters, Modout learns the probabilities of connecting the units belonging to different modalities.

In Modout, given N_m modalities, the stochastic mask M_j for layer j is defined as

$$M_j = C_j U_j C_{j-1}^T \tag{1}$$

where C_j is a $N_j \times N_m$ binary matrix and U_j is a $N_m \times N_m$ binary matrix, $U_j \sim Bernoulli(P_j)$. In our work, the modality-wise probability matrix P is trained together with the rest of the network parameters. However, the diagonal elements of P are fixed to unity in order to guarantee that all the signals for a modality can be passed to the units which belong to that modality in the next layer. We note that the number of additional parameters to learn for a mask is only



(a) independent (b) fused (c) fully-connected

Fig. 1: Three typical fusion architectures achievable by Modout, and their corresponding weight masks.

$N_m(N_m-1)$, which is significantly less than Blockout which requires learning K probabilities for each unit.

To learn the probability matrix P, ideally we want to update it via gradient descent like the other network parameters. However, the gradient of P is not available because P is related to the cost function by sampling, which is not differentiable. [12] addressed a similar problem when attempting to compute the gradient of the loss with respect to the cluster probabilities parameterizing the cluster assignments. Here, they simply used the gradient of the loss with respect to the cluster assignments, masked by the assignment matrix such that the gradient of unselected clusters is zero. We use a similar method to learn the gradients of the probability matrix.

4 Experiments

In this section, we investigate the ability of Modout to learn a strategy that is competitive with a hand-designed fusion structure [3] on a large-scale gesture recognition benchmark.

4.1 Montalbano gesture recognition dataset

The Montalbano dataset, originally released as part of the Chalearn 2014 Looking at People Challenge (track 3) [13], is used for evaluating our approach. It consists of 13,858 instances of Italian conversational gestures performed by different people and recorded with a consumer RGB-D sensor. It includes color, depth video and mocap (articulated pose) streams. The gestures are drawn from a large vocabulary, from which 20 categories are identified to be detected and recognized, while the rest are considered as arbitrary movements. Each gesture in the training set is accompanied by a ground truth label as well as information about its start-and-end points. An additional audio modality not in the 2014 competition was re-synced and added in [3], which we also consider.

4.2 Network Architecture and experimental setup

The network architectures for testing are based on the one in [3]. In that work, each modality is pre-trained as an individual classifier (video modalities use two-stage convolutional networks, the audio stream uses a one-stage convolutional network, and the mocap stream uses a MLP with two hidden layers). The penultimate layer of each modality-specific classifier is then connected via a shared hidden layer to a softmax output and the whole system is then trained by a two-stage procedure. First, the weights to and from the shared layer are initialized such that the overall network performs a simple fusion of modalities. Then, this constraint is gradually relaxed to permit a more flexible fusion strategy. In addition to this careful initialization and relaxation process, prior knowledge influences the *order* in which modalities are fused. The depth and intensity channels corresponding to each hand are fused, while cross-modality fusion involving the other channels are postponed until the shared layer. This network architecture achieved first place out of 17 teams in the ChaLearn 2014 Looking at People Challenge (gesture recognition track) [13].

Two experiments are conducted to evaluate the performance of Modout. The first experiment aims to compare the performance of Modout with other stochastic regularization methods using a simple network architecture, namely a MLP. Due to the high dimensionality of the raw data and limited number of training sequences, directly applying a MLP to the raw data leads to poor generalization. Instead, we use the intermediate outputs from the first fully-connected layer of the pre-trained classifiers as the input features for each modality. The number of total input features is 2600, including 800 audio features, 900 mocap features, 450 color features, and 450 depth features. Color and depth features have been concatenated into a single video modality. The methods we include for comparison include standard (non-regularized) backpropagation, Dropout, Blockout, ModDrop, Modout, and the combination of Modout and Dropout. A MLP with two hidden layers followed by a softmax regression layer is used for all the tests. The number of units in each hidden layer is set to 3,000. This experiment is only performed on the data released early in the challenge. We apply the standard practice of removing frames with no gesture present during preprocessing. The frames are divided into training data, validation data, and testing data using the same split as in [3].

The second experiment aims to evaluate the performance of Modout by integrating it into the network architecture in [3]. This is done by concatenating each of the last two layers for each modality into a single layer and adding connections which are modulated by Modout. Thus, the network structure becomes a MLP with two hidden layers and one softmax regression layer. The real test data released later in the challenge are used for testing. Similar to [3], we first use a motion detector to remove the frames without motion in the test data, and then use the learned model to classify all the frames. The Jaccard index is used to measure performance:

$$J_{s,n} = \frac{|A_{s,n} \cap B_{s,n}|}{|A_{s,n} \cup B_{s,n}|} \tag{2}$$

where $A_{s,n}$ and $B_{s,n}$ denote the binary ground truth and predictions for gesture category n and sequence s respectively. The final score is measured by the mean Jaccard index over all categories and sequences.

The input to the network is a so-called *dynamic pose* consisting of synchronized multi-modal measurements concatenated from several frames temporally spaced with a given stride. In [3], different networks are trained on different strides and results are combined. In our experiments, we only use a single temporal stride of 4 for sampling for all the modalities, as the result is very close to using a combination of strides as also shown in [3].

4.3 Experimental results and analysis

Table 1 shows the result of different stochastic regularization methods using the pre-trained intermediate representation of each modality as input. The drop-out rate is set to 0.2 for the first hidden layer and 0.5 for the second hidden layer. The number of clusters in Blockout is set to 2 which was found empirically to have the best performance. For both Blockout and Modout, the probabilities are initialized to 0.5 in order to maximize uncertainty at the beginning of training. The results show that there is no significant difference between Dropout and ModDrop, while Modout and its combination with Dropout significantly outperform the other methods which ignore modality information. In this first experiment we report performance on classification of dynamic poses (as opposed to gesture localization), therefore the metric used is classification accuracy. The best test accuracy is achieved by a combination of Modout and Dropout.

Table 1: Classification accuracy(%) of different stochastic regularization methods.

	Validation accuracy	Test accuracy
BackProp	91.1	92.0
Dropout	91.5	92.5
Blockout	91.7	92.6
ModDrop	92.1	92.4
Modout	91.6	93.6
Modout+Dropout	**92.9**	**93.8**

Table 2 gives results on full gesture detection and localization reported as Jaccard Index (Eq. 2). It shows that our approach (Modout + Dropout) achieves a score of 0.888, which is higher than [3] using either Dropout or ModDrop + Dropout on a similar

but carefully chosen fusion architecture. Compared to previously reported results, our result is only eclipsed by the state-of-the-art, [14] which uses a combination of temporal convolution layers and Long Short-Term Memory (LSTM). One possible reason is that the temporal correlation between two adjacent spatio-temporal blocks is not considered in our approach. Our result could be further improved by using a simple 1-D Markov Random Field model as a post-processing step.

Table 2: Comparison with state-of-the-art results recently published for the same task.

Approach	Jaccard index
Wu et al. (2016) (HMM, DBM, 3DCNN) [15]	0.809
Chang (2014) (MRF, KNN, HoG) [16]	0.827
Monnier et al. (2014) (AdaBoost, HoG) [17]	0.834
Neverova et al. (2016) (Dropout) [3]	0.876
Neverova et al. (2016) (ModDrop + Dropout) [3]	0.880
Pigou et al. (2016) (Temp Conv + LSTM) [14]	**0.906**
Ours (Modout + Dropout)	0.888

4.4 Comparing Modout to early fusion and late fusion

Early fusion and late fusion are the two most common fusion strategies. They are actually two extremes of Modout, i.e., when the off-diagonal elements of the probability matrix are set to $p_{i,j} = 1$ and $p_{i,j} = 0$ respectively. In this section, we show that by effectively learning the probability of fusing modalities in each layer, Modout can learn a structure that outperforms both early fusion and late fusion.

To validate this assumption, experiments are performed on two datasets. The first dataset is a simulated multimodal dataset using the well-known MNIST handwritten digit database. The image is split into four segments similar to ModDrop, each segment representing one modality. The second dataset is the Montalbano dataset considered above. Similar to the previous experiment, intermediate features of three modalities including video, skeleton, and audio are used. The network is first trained with Modout. After the probabilities are learned, a weight mask is created by binarizing the probabilities, and the model is trained again using the weight masks in a deterministic manner.

Table 3: Comparison with early fusion and late fusion (error rates in percentages).

	MNIST	Montalbano
Modout	**1.03**	6.44
Early fusion	1.19	7.23
Late fusion	1.88	6.94
Re-trained using learned structure	1.04	**6.01**

The result is shown in Table 3. We see that early fusion is better than late fusion for the MNIST dataset, while late fusion is better for the Montalbano dataset. The rationale behind is that the modalities of the MNIST dataset are highly correlated because they are from the same image, while the audio, video, and skeleton modalities in the Montalbano dataset have less correlation. For the case of Montalbano, we also see that binarizing the probabilities and fine-tuning deterministically learns a superior model.

5 Conclusions

We have presented Modout, an extension of ModDrop that is particularly useful for multi-modal learning. It can be applied to multiple layers, and has the capability of learning modality fusion. While motivated by the challenge of learning fusion structure, Modout can leverage any known grouping of the inputs.

We presented experimental results on a challenging multimodal dataset, which shows that Modout outperforms other stochastic regularization methods, and achieves close to the state-of-the-art for gesture recognition. Also, pruned network structure using the probabilities learned by Modout performs better than both early fusion and late fusion. Future work includes applying our approach to other types of neural network structures and validating on other multimodal datasets.

References

[1] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011, pp. 689–696.

[2] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in Neural Information Processing Systems (NIPS) 25*, 2012, pp. 2222–2230.

[3] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, "Moddrop: adaptive multi-modal gesture recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2016, to appear.

[4] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, "Deep multimodal feature analysis for action recognition in RGB+D videos," *arXiv preprint arXiv:1603.07120*, 2016.

[5] Z. Wang, R. Lin, J. Lu, J. Feng *et al.*, "Correlated and individual multi-modal deep learning for RGB-D object recognition," *arXiv preprint arXiv:1604.01655*, 2016.

[6] A. Wang, J. Cai, J. Lu, and T.-J. Cham, "MMSS: Multi-modal sharable and specific feature learning for RGB-D object recognition," in *Proceedings of the International Conference on Computer Vision*, 2015, pp. 1125–1133.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] L. Wan, M. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using DropConnect," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013, pp. 1058–1066.

[9] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, "Deep networks with stochastic depth," *arXiv preprint arXiv:1603.09382*, 2016.

[10] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 648–656.

[11] P. Kulkarni, J. Zepeda, F. Jurie, P. Pérez, and L. Chevallier, "Learning the structure of deep architectures using L1 regularization," in *Proceedings of the British Machine Vision Conference*, 2015.

[12] C. Murdock, Z. Li, H. Zhou, and T. Duerig, "Blockout: Dynamic model selection for hierarchical deep networks," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *ECCV 2014 Workshops*. Springer, 2014, pp. 459–473.

[14] L. Pigou, A. v. d. Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *arXiv preprint arXiv:1506.01911*, 2015.

[15] D. Wu, L. Pigou, P.-J. Kindermans, N. Le, L. Shao, J. Dambre, and J.-M. Odobez, "Deep dynamic neural networks for multimodal gesture segmentation and recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2016, to appear.

[16] J. Y. Chang, "Nonparametric gesture labeling from multi-modal data," in *Computer Vision-ECCV 2014 Workshops*, 2014, pp. 503–517.

[17] C. Monnier, S. German, and A. Ost, "A multi-scale boosted detector for efficient and robust gesture recognition," in *ECCV 2014 Workshops*, 2014, pp. 491–502.